IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727, Volume 7, Issue 6 (Nov. - Dec. 2012), PP 04-12
www.iosrjournals.org

Privacy Preserving Naïve Bayes Classifier for Horizontally Distribution Scenario Using Un-trusted Third Party

Alka Gangrade[1], Ravindra Patel[2]
[1](*T.I.T.-M.C.A., Technocrats Institute of Technology, Bhopal, India)*
[2](Dept. of M.C.A., U.I.T. R.G.P.V., Bhopal, India)

Abstract : *The aim of the classification task is to discover some kind of relationship between the input attributes and the output class, so that the discovered knowledge can be used to predict the class of a new unknown tuple. The problem of secure distributed classification is an important one. In many situations, data is split between multiple organizations. These organizations may want to utilize all of the data to create more accurate predictive models or classifier while revealing neither their training data nor the tuples to be classified. The Naïve Bayes classifier is a simple but efficient baseline classifier. In this paper, we present a privacy preserving Naïve Bayes classifier for horizontally partitioned data. Our three layer protocol uses an Un-trusted Third Party (UTP). We study how to calculate model parameters for privacy preserving three-layer Naïve Bayes classifier for horizontally partitioned databases and communicate their intermediate results to the UTP not their private data. In our protocol, an UTP allows to meet privacy constraints and achieve acceptable performance.*

Keywords: *Naïve Bayes classifier, Privacy preserving, Probability, Un-trusted Third Party (UTP).*

I. Introduction

Nowadays Privacy preserving data mining is the most challenging research area within the data mining society. In several cases, multiple parties may wish to share aggregate private data, without leaking any sensitive information at their end [1]. This requires secure protocols for sharing the information across the different parties. The data may be distributed in two ways: Horizontal partitioned data and Vertical partitioned data. Horizontal partition means, where different sites have different sets of records containing the same attributes. Vertical partition means, where different sites have different attributes of the same sets of records [2].

The objective of privacy preserving data classification is to build accurate classifiers without disclosing private information in the data being mined. For example, a classifier capable of identifying risky loans could be used to aid in the decision of whether to grant a loan to an individual. In this paper, we particularly focus on privacy preserving Naïve Bayes classification on horizontally partitioned data using UTP.

1.1 Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem. Studies comparing classification algorithm have found a simple Bayesian classifier known as the Naïve Bayesian classifier to be comparable in performance with decision tree and selected neural network classifier. Bayesian classifiers have also exhibited high accuracy and speed when applied to large database.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve". Next section reviews basic probability notation and Bayes' Theorem from Han and Kamber [3].

1.1.1 Bayes' Theorem

Bayes' theorem is named after Thomas Bayes, a nonconformist English Clergyman, who did early work in probability and decision theory during the 18th century. Let X is a data tuple. In Bayesian terms, X is considered "evidence". As usual, it is described by measurements made on a set of n attributes. Let H is some hypothesis, such as that the data tuple X belongs to a specified class c. For classification problems, we want to determine P(H|X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X.
P(H|X) is the posterior probability of H conditioned on X.
P(H) is the prior probability of H. For our example; this is the probability that any given customer will buy a computer, regardless of age, income or any other information. P(X|H) is the posterior probability of X conditioned on H.

Bayes' theorem is useful in that it provides a way of calculating the posterior probability P(H|X), from P(H), P(X|H) and P(X). Bayes' theorem is

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)} \quad (1)$$

1.2 Privacy preserving Naïve Bayes classifier for horizontally partitioned data

In this section, we will focus on secure Naïve Bayesian classifier on horizontally partitioned data. Records of different patients that are treated by different hospitals can be seen as an example of horizontally partitioned data. All of the information for a given patient is contained at one hospital, but different hospitals have different patients.

In order to see how a privacy preserving Naïve Bayesian classifier is constructed, we need to address two issues: how to calculate the probability for each attribute and how to classify a new tuple [4, 5, 6]. The following subsections provide details on both issues. The protocols presented below are very efficient. However, they compromise a little on security. At the end of the protocol, all parties learn only model parameters not the attribute value. We present a protocol which does not reveal anything except the final classifier or model parameters.

For horizontally partitioned data model, all the attributes needed for classifying new tuples are held by one party. Therefore, given the probabilities, no collaboration is needed for classifying a new tuple. At the first look it appears that a more secure solution can be obtained by hiding the probabilities and using a secure protocol to classify each new tuple. Since each party has knowledge of all of the attributes, eventually by classifying sufficiently many tuples, a party may learn or understand the hidden model or counts quite easily. Therefore, we do not try to hide the model in the horizontally partitioned data through Naïve Bayes classification method. Instead, we simply try to learn the Naïve Bayes classifier without revealing anything else.

1.3 Secure protocols with Un-trusted Third Party (UTP)

A straightforward solution for privacy preserving data mining is to use a trusted third party to gather data from all data sources and then send back results after running the desired data mining process. However, the level of trust is not acceptable in this scheme since the privacy of the data sources cannot be protected from the third party. There have been several approaches to support privacy preserving data mining over multi-party without using third parties [7, 8]. The existence of an UTP enables efficient protocols without revealing private information. The idea of an UTP is that it is willing to perform some computation for the parties in the protocol. It is not trusted with the data or the results. The trust placed in this party is that it does not join with any of the participating parties to violate information privacy and correctly executes the protocol. Correct execution of the protocol is only required to guarantee correct result, even a dishonest third party is unable to learn private information in the absence of collusion. Typically the third party is given some information in the form of intermediate result. We simply mean that the third party cannot make any sense of the data given to it without the assistance of the local parties involved in the protocol. The third party performs a computation on the intermediate result, possibly exchanging information with the other parties in the process.

1.4 Our Contributions

Our main contributions in this paper are as follows:

- We present a novel three-layer privacy preserving Naïve Bayes classifier.
- It proposes a new protocol to calculate model parameters for horizontally partitioned databases.
- It classifies new tuple by using model parameters.

1.5 Organization of the paper

The rest of the paper is organized as follows. In Section 2, we discuss the related work. Section 3, describes our three-layer privacy preserving Naïve Bayes classification model for horizontally partitioned data. Section 3.1 describes architecture of our model. Section 3.2 sets some assumptions. Section 3.3 and 3.4 describe informal algorithm and formal algorithms of our proposed work respectively. In Section 4, we present our calculation and results that are conducted by using our privacy preserving three-layer Naïve Bayes classifier on real-world data sets. In Section 5, we conclude our paper with the discussion of the future work.

II. Related Work

Privacy preserving data mining has been an active research area for a decade. A lot of work is going on by the researcher on privacy preserving classification in distributed data mining. The first Secure Multiparty Computation (SMC) problem was described by Yao [9]. SMC allows parties with similar background to compute result upon their private data, minimizing the threat of disclosure was explained [10]. A variety of tools discussed and how they can be used to solve several privacy preserving data mining problem [11]. We now give some of the related work in this area. Preserving customer privacy by distorting the data values proposed by Agrawal and Srikant [8]. Since then, there has been work improving this approach in various ways [12].

Classification is one of the most widespread data mining problems come across in real life. General classification techniques have been extensively studied for over twenty years. The classifier is usually represented by classification rules, decision trees, Naïve Bayes classification, neural networks. First ID3 decision tree classification algorithm is proposed by Quinlan [13]. Lindell and Pinkas proposed a secure algorithm to build a decision tree using ID3 over horizontally partitioned data between two parties using SMC [14]. A novel privacy preserving distributed decision tree learning algorithm [15] that is based on Shamir [16]. The ID3 algorithm is scalable in terms of computation and communication cost, and therefore it can be run even when there is a large number of parties involved and eliminate the need for third party and propose a new method without using third parties. A generalized privacy preserving variant of the ID3 algorithm for vertically partitioned data distributed over two or more parties introduced in [17, 18, 19, 20] and horizontally partitioned data distributed over multi parties introduced in [21, 22]. Privacy preserving Naïve Bayes classification for horizontally partitioned data introduced in [4] and vertically partitioned data introduced in [5, 6]. Centralized Naïve Bayes classification probability calculation is introduced in [23].

III. Proposed Work

In this paper, we address the issue related to privacy preserving data mining in distributed environment. In particular, we focus on privacy preserving three-layer Naïve Bayes classification for horizontally partitioned databases using UTP. The objective of privacy preserving data classification is to build accurate classifiers without disclosing private information in the data being mined. The performance of privacy preserving techniques should be analyzed and compared in terms of both the privacy protection of individual data and the predictive accuracy of the constructed classifiers.

3.1 Architecture

Proposed architecture of our privacy preserving three-layer Naïve Bayes Classifier for horizontally partitioned databases is shown in Fig.1.

3.1.1 Input Layer – Input layer comprises of all the parties that are involved in the classification process. They individually calculate all counts such as total number of tuples, counts for each class value of every attribute value for each attribute and send all these counts as an intermediate result form to UTP.

3.1.2 Computation Layer – The UTP exists at the 2nd layer i.e. the computation layer of our protocol. UTP collects only intermediate results from all parties not data and calculate total probability for each class value of every attribute value for each attribute from the intermediate result. Finally send all these probabilities or model parameters to all the parties for further calculation.

3.1.3 Output Layer – In horizontally partitioned data, each party has all the attributes. Each party receives all probabilities or model parameters from computation layer. Each party is able to classify the new tuple in same manner. There is no need to hide the tuple. No collaboration is needed for classifying a new tuple and protocol is secure to classify each new tuple.

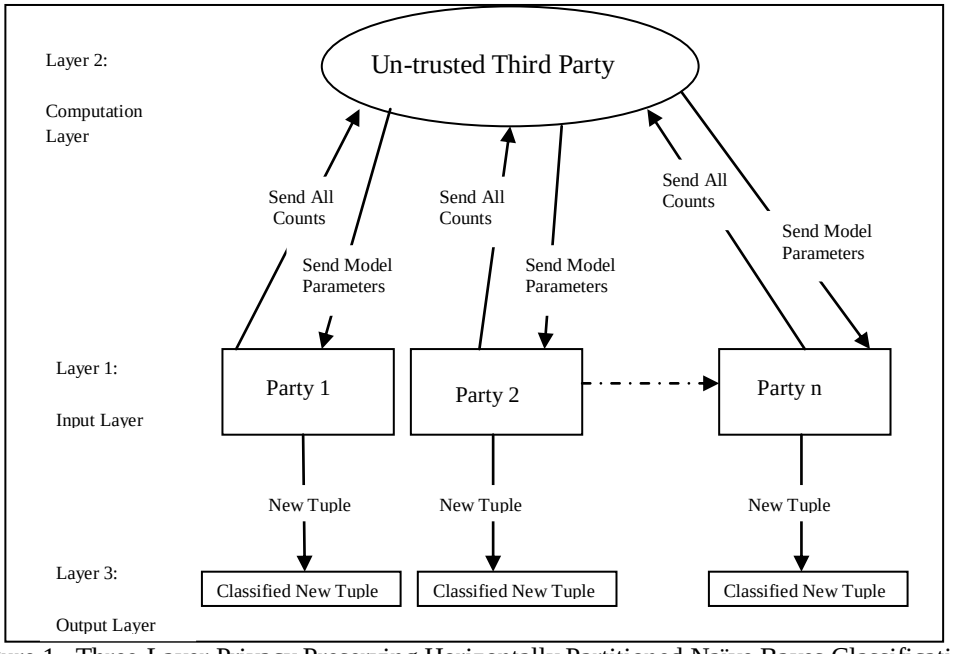

Figure 1. Three-Layer Privacy Preserving Horizontally Partitioned Naïve Bayes Classification Model

3.2 Assumptions

The following assumptions have been set:

- UTP computes total probabilities from the intermediate results provided by all parties.
- UTP sends total probabilities to all party.
- Every party separately classify new tuple, therefore security is not needed for horizontally partitioned data at this level.
- Each party is not communicating their input data to other party.
- The communication networks used by the input parties to communicate with the UTP are secure.

3.3 Informal Algorithm

Building the classifier model for horizontal partition of data, each party has partial information about every attribute. Each party can locally compute the local count of tuples. UTP securely calculates the total or global counts by the sum of the local counts. The required probability can be computed by dividing the appropriate global sums. The procedure for calculating the parameters for nominal attributes is described below.

3.3.1 Input Layer

- Party individually calculates the counts for each value of every attribute for every class value and send to the UTP.
- Party individually calculates total number of tuples they have and send to the UTP.

3.3.2 Computation Layer

- UTP computes the total counts of all parties of each value of every attribute for every class value.
- UTP computes the probabilities of each value of every attributes for every class value.
- UTP sends these probabilities (model parameters) to all party for classify new tuples.
-

3.3.3 Output Layer

- Any party can classify the new tuple by using probabilities (model parameters).
-

3.4 Formal Algorithms

Require:

1. n parties i.e. P_1, P_2, .., P_n,
2. c class values i.e. c_1, c_2, .., c_c,
3. a attribute name A_1, A_2, .., A_a, A_c, where A_c is the class attribute.

Note:

1. C^p_{xyz} —> represents number of tuples with party Pp having class z, attribute value y of attribute Ax.
2. A_{xy} —> represents attribute name A_x with y attribute values.
3. N^p_z —> represents number of tuples with party Pp having class z.
4. T —> represents total tuples with all participating parties.
5. $Prob_{xyz}$ —> represents probability of attribute A_x with attribute value y having class z.
6. $Prob_z$ —> represents probability of class z.
7. C_{xyz} —> represents number of tuples with all participating parties having class z, attribute value y of attribute A_x.
8. N_z —> represents number of tuples with all participating parties having class z.
9. New.A_{xy} —>represents attribute name A_x with attribute value y of new tuple to be classified.

Algorithm 1: 3LPPHPNBC ( ) – Three-layer privacy preserving horizontally partitioned NBC.

1. Local_Att_Count ( )
2. Local_Class_Count ( )
3. Global_Att_Count ( )
4. Global_Class_Count ( )
5. Cal_Att_Prob ( )
6. Classify_Tuple ( ) { Classify tuple by calling Cal_Total_Prob (c_z) }

3.4.1 Input Layer

Algorithm 2: Local_Att_Count ( ) - Calculate local counts for each party for each attribute value for every attribute for all class value.

1. For Party P_i where i = 1 to n do
2. For Attribute A_j where j = 1 to a do
3. For Attribute value y_k where k = 1 to y_j do

4. For Class value c_z where z = 1 to c do
   i. $C^i_{jkz} = 0$
5. For all tuples having class value z
   i. $C^i_{jkz} = C^i_{jkz} + 1$
6. End for
7. End for
8. End for
9. End for
10. End for.

Algorithm 3: Local_Class_Count ( ) - Calculate local counts for each party for all class value and total number of tuples with all participating parties.

1. T = 0
2. For Party P_i where i = 1 to n do
3. For Class value c_z where z = 1 to c do
   i. $N^i_z = 0$
4. For all tuples having class value z
   i. T = T + 1
   ii. $N^i_z = N^i_z + 1$
5. End for
6. End for
7. End for.

3.4.2 Computation Layer

Algorithm 4: Global_Att_Count ( ) - Calculate total or global counts for each attribute value for every attribute for all class value for all participating parties.

1. For Class value c_z where z = 1 to c do
2. For Attribute A_j where j = 1 to a do
3. For Attribute value y_k where k = 1 to y_j do
   i. $C_{jkz} = 0$
4. For Party P_i where i = 1 to n do
   i. $C_{jkz} = C_{jkz} + C^i_{jkz}$
5. End for
6. End for
7. End for
8. End for.

Algorithm 5: Global_Class_Count ( ) - Calculate total or global counts and probabilities for all class value.

1. For Class value c_z where z = 1 to c do
   i. $N_z = 0$
2. For Party P_i where i = 1 to n do
   i. $N_z = N_z + N^i_z$
3. End for
4. $Prob_z = N_z / T$
5. End for.

Algorithm 6: Cal_Att_Prob ( ) - Calculate Probability of each attribute value of every attribute for all class value.

1. For Attribute A_j where j = 1 to a do
2. For Attribute value y_k where k = 1 to y_j do
3. For Class value c_z where z = 1 to c do
   i. $Prob_{jkz} = C_{jkz} / N_z$
4. End for
5. End for
6. End for

3.4.3 Output Layer: Algorithms for classifying new tuple.

Algorithm 7: Classify_Tuple ( ) – Find class having maximum total probability and classify tuple.

1. Max_Prob = 0
2. Class = Null
3. For Class value c_z where z = 1 to c do
4. Prob = Cal_Total_Prob (c_z)

5. If Prob > Max_Prob then
   i. Max_Prob = Prob
   ii. Class = c_z
6. End if
7. End for
8. Return Class.

Algorithm 8: Cal_Total_Prob (c_z) - Calculate total Probability for all class value of new tuple.

1. For Class value c_z where z = 1 to c do
   i. $Total\_Prob_z$ = 1
2. For Attribute A_j where j = 1 to a do
3. For Attribute value y_k where k = 1 to y_j do
4. If A_{jk} = New.A_{jv} then
   i. $Total\_Prob_z$ = $Total\_Prob_z$ * $Prob_{jkz}$
   ii. Break
5. End if
6. End for
7. End for
   i. $Total\_Prob_z$ = $Total\_Prob_z$ * $Prob_z$
   ii. Return $Total\_Prob_z$
8. End for.

IV. Evaluation And Result

Since all the model parameters are completely present with all the parties, evaluation is simple. The party that wants to evaluate the tuple simply uses the Naïve Bayes evaluation procedure locally to classify the tuple. The other parties have no interaction in the process. Thus, there is no question of privacy being compromised.

Table 1: Party P_1

Rid	Age	Income	Student	Credit_rating	Class : Buys_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31..40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31..40	Low	Yes	Excellent	Yes

Table 2: Party P_2

Rid	Age	Income	Student	Credit_rating	Class : Buys_computer
1	<=30	Medium	No	Fair	No
2	<=30	Low	Yes	Fair	Yes
3	>40	Medium	Yes	Fair	Yes
4	<=30	Medium	Yes	Excellent	Yes
5	31..40	Medium	No	Excellent	Yes
6	31..40	High	Yes	Fair	Yes
7	>40	Medium	No	Excellent	No

Probabilities calculation of the attributes of horizontally partitioned databases:

For Party P_1

Total Number of tuples = 7

Class Y: Buys_computer = "Yes" Total tuples for Class Y = 4

Class N: Buys_computer = "No" Total tuples for Class N = 3

For Party P_2

Total Number of tuples = 7

Class Y: Buys_computer = "Yes" Total tuples for Class Y = 5

Class N: Buys_computer = "No" Total tuples for Class N = 2

Table 3: Compute probability for Age

Age	Class Y				Class N			
	Party P_1	Party P_2	Total	Probability	Party P_1	Party P_2	Total	Probability
<=30	0	2	2	0.2222	2	1	3	0.6
31..40	2	2	4	0.4444	0	0	0	0.0
>40	2	1	3	0.3333	1	1	2	0.4

Table 4: Compute probability for Income

Income	Class Y				Class N			
	Party P_1	Party P_2	Total	Probability	Party P_1	Party P_2	Total	Probability
High	1	1	2	0.2222	2	0	2	0.4
Medium	1	3	4	0.4444	0	2	2	0.4
Low	2	1	3	0.3333	1	0	1	0.2

Table 5: Compute probability for Student

Student	Class Y				Class N			
	Party P_1	Party P_2	Total	Probability	Party P_1	Party P_2	Total	Probability
Yes	2	4	6	0.6667	1	0	1	0.2
No	2	1	3	0.3333	2	2	4	0.8

Table 6: Compute probability for Cradit_rating

Cradit_ rating	Class Y				Class N			
	Party P_1	Party P_2	Total	Probability	Party P_1	Party P_2	Total	Probability
Fair	3	3	6	0.6667	1	1	2	0.4
Excellent	1	2	3	0.3333	2	1	3	0.6

Table 7: Compute probability for Class

Class : Buys_co mputer	Class Y				Class N			
	Party P_1	Party P_2	Total	Probability	Party P_1	Party P_2	Total	Probability
	4	5	9	0.6429	3	2	5	0.3571

Table 8: Classify new tuples of Party P_1

Rid	Age	Income	Student	Credit_rating	Class : Buys_computer
8	<=30	High	Yes	Fair	?
9	>40	High	No	Excellent	?
.	.	.	.	.	.

For Rid =8
Likelihood of the two classes:
For Class Y = 0.2222 * 0.2222 * 0.6667 * 0.6667 * 0.6429 = 0.01411
For Class N = 0.6 * 0.4 * 0.2 * 0.4 * 0.3571 = 0.00686
Conversion into a probability by normalization
P (Class Y) = 0.01411/ (0.01411 + 0.00686) = 0.673
P (Class N) = 0.00686/ (0.01411 + 0.00686) = 0.327
Here P(Class Y) > P(Class N) then Class : Buys_computer = Yes
For Rid =9
Likelihood of the two classes:
For Class Y = 0.3333 * 0.2222 * 0.3333 * 0.3333 * 0.6429 = 0.00529
For Class N = 0.4 * 0.4 * 0.8 * 0.6 * 0.3571 = 0.02743
Conversion into a probability by normalization
P (Class Y) = 0.00529/ (0.00529 + 0.02743) = 0.162
P (Class N) = 0.02743/ (0.00529 + 0.02743) = 0.838
Here P(Class Y) < P(Class N) then Class : Buys_computer = No

Table 9: Classify new tuples of Party P_2

Rid	Age	Income	Student	Credit_rating	Class : Buys_computer
8	<=30	Medium	Yes	Excellent	?
9	31..40	Low	No	Fair	?

For Rid =8
Likelihood of the two classes:
For Class Y = 0.2222 * 0.4444 * 0.6667 * 0.3333 * 0.6429 = 0.01411
For Class N = 0.6 * 0.4 * 0.2 * 0.6 * 0.3571 = 0.01029
Conversion into a probability by normalization
P (Class Y) = 0.01411/ (0.01411 + 0.01029) = 0.578
P (Class N) = 0.01029/ (0.01411 + 0.01029) = 0.422
Here P(Class Y) > P(Class N) then Class : Buys_computer = Yes
For Rid =9
Likelihood of the two classes:
For Class Y = 0.4444 * 0.3333 * 0.3333 * 0.6667 * 0.6429 = 0.02116
For Class N = 0.0 * 0.2 * 0.8 * 0.4 * 0.3571 = 0.0
Conversion into a probability by normalization
P (Class Y) = 0.02116/ (0.02116 + 0.0) = 1.0
P (Class N) = 0.0/ (0.02116 + 0.0) = 0.0
Here P(Class Y) > P(Class N) then Class : Buys_computer = Yes

Table 10: Execution time comparison for Computing Model Parameters

S.No.	Number of Tuples	NBC (ms)	3LPPHPNBC (ms)
1	14	70	20
2	25	83	32
3	50	99	42
4	100	112	50
5	200	135	57

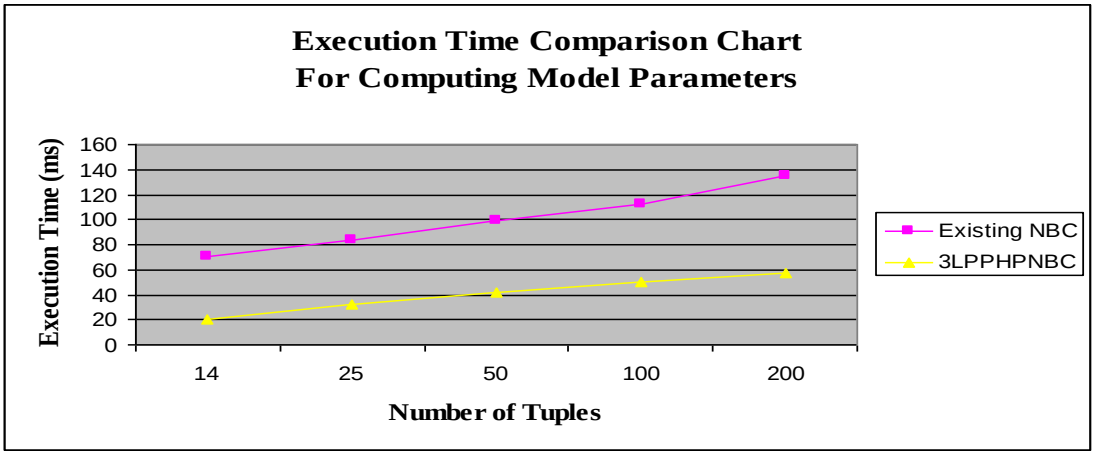

Figure 2. Execution time comparison chart

Estimated time for classifying a new tuple is 12 ms.

Table 11: 3LPPHPNBC Accuracy

S.No.	Number of Tuples	Accuracy
1	14	78.57%
2	25	80.%
3	50	82%
4	100	83%
5	200	84%

We are running our algorithms on 50% training data and 50% test data. We find that execution time for calculating model parameters is less than the existing Naïve Bayes classifier, it is shown by fig 2 and the accuracy is almost same, it is given in Table 11.

V. Conclusion And Future Work

We believe that it is possible to produce a privacy preserving Naïve Bayes classifier for horizontally partitioned databases with SMC techniques. In this paper, we proposed a new classifier using three-layer

architecture that enables SMC by hiding the identity of the parties taking part in the classification process using UTP. Further we may describe that intermediate result is calculated by every party individually and send only intermediate result to UTP not the input data. Through the communication between UTP and all party, final result is carried out. It requires less memory space and provides fast and easy calculations. Using this protocol, data will almost secure and privacy of individual will be maintained. Further development of the protocol is expected for joining multi-party using Trusted Third Party (TTP). We are continuing work in this field to develop Naïve Bayes classifier for vertically partitioned databases and also analysis new as well as existing classifiers.

Acknowledgements

We are grateful to the University and the College for their support. We express gratitude to my colleagues for their technical support and the referees for their beneficial suggestions.

References

[1] B. Pinkas, Cryptographic techniques for privacy-preserving data mining, *ACM SIGKDD Explorations Newsletter, 4(2)*, 2006, 12-19.

[2] C. C. Aggarwal, P. S. Yu., *Privacy-Preserving Data Mining:Models and Algorithms*(London, Kluwer Academic Publishers Boston).

[3] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*(India, Elsevier).

[4] M. Kantarcioglu and J. Vaidya, Privacy preserving naive Bayes classifier for horizontally partitioned data, *In IEEE ICDM Workshop on Privacy Preserving Data Mining*, Melbourne, FL, November 2003, 3-9.

[5] J. Vaidya and C. Clifton, Privacy preserving naive Bayes classifier on vertically partitioned data, *Proc. SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, April 2004, 22-24.

[6] Z. Yang and R. Wright, Privacy-Preserving Computation of Bayesian Networks on Vertically Partitioned Data, *IEEE Transactions on Data Knowledge Engineering, 18(9*), April 2006, 1253-1264.

[7] V. Verykios, E. Bertino, State-of-the-art in Privacy preserving Data Mining, *SIGMOD Record, 33(1)*, 2004, 50-57.

[8] R. Agrawal, R. Srikant Privacy preserving data mining, *Proc. of the ACM SIGMOD on Management of data*, Dallas, TX USA, May 15-18, 2000, 439-450.

[9] A. C. Yao, Protocols for secure computation, *Proc. of 23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 1982, 160-164.

[10] W. Du, Mikhail J. Attalah, Secure multi-problem computation problems and their applications: A review and open problems, *Tech. Report CERIAS Tech Report 2001-51*, Center for Education and Research in Information Assurance and Security and Department of Computer Sciences, Purdue University, West Lafayette, IN 47906, 2001.

[11] C. Cliffton, M. Kantarcioglu, J. Vaidya, Tools for privacy preserving distributed data mining, *ACM SIGKDD Explorations Newsletter, 4(2)*, 2004, 28-34.

[12] D. Agrawal and C. C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, *Proc. of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Santa Barbara, California, USA, May 21-23 2001, 247–255.

[13] J. R. Quinlan, Induction of decision trees, i*n: Jude W. Shavlik, Thomas G. Dietterich, (Eds.), Readings in Machine Learning, Morgan Kaufmann, 1*, 1990, 81–106.

[14] Y. Lindell, B. Pinkas, Privacy preserving data mining, *Journal of Cryptology, 15(3)*, 2002, 177–206.

[15] F. Emekci , O. D. Sahin, D. Agrawal, A. El Abbadi, Privacy preserving decision tree learning over multiple parties, *Data & Knowledge Engineering 63*, 2007, 348-361.

[16] A. Shamir, How to share a secret, *Communications of the ACM, 22(11)*, 1979, 612-613.

[17] W. Du, Z. Zhan, Building decision tree classifier on private data, *In CRPITS*, 2002, 1–8.

[18] J. Vaidya, C. Clifton, M. Kantarcioglu, A. S. Patterson, Privacy-preserving decision trees over vertically partitioned data, *Proc. of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, 2008, 139–152.

[19] J. Shrikant Vaidya, *Privacy preserving data mining over vertically partitioned data*, doctoral diss., Purdue University, August 2004.

[20] W. Fang, B. Yang, Privacy Preserving Decision Tree Learning Over Vertically Partitioned Data, *Proc. of the 2008 International Conference on Computer Science & Software Engineering*, 2008, 1049-1052.

[21] A. Gangrade, R. Patel, A novel protocol for privacy preserving decision tree over horizontally partitioned data, *International Journal of Advanced Research in Computer Science, 2 (1)*, 2011, 305-309.

[22] A. Gangrade, R. Patel, Privacy Preserving Two-Layer Decision Tree Classifier for Multiparty Databases, *International Journal of Computer and Information Technology (2277 – 0764), 1(1)*, 2012, 77-82.

[23] I. H. Witten, E. Frank and M. A. Hall, *Data Mining Practical Machine Learning Tools and Techniques* (Burlington, MA : Morgan Kaufmann, 2011).